

Mayıs, 2024

# SQL PROJESİ-KODLASAM

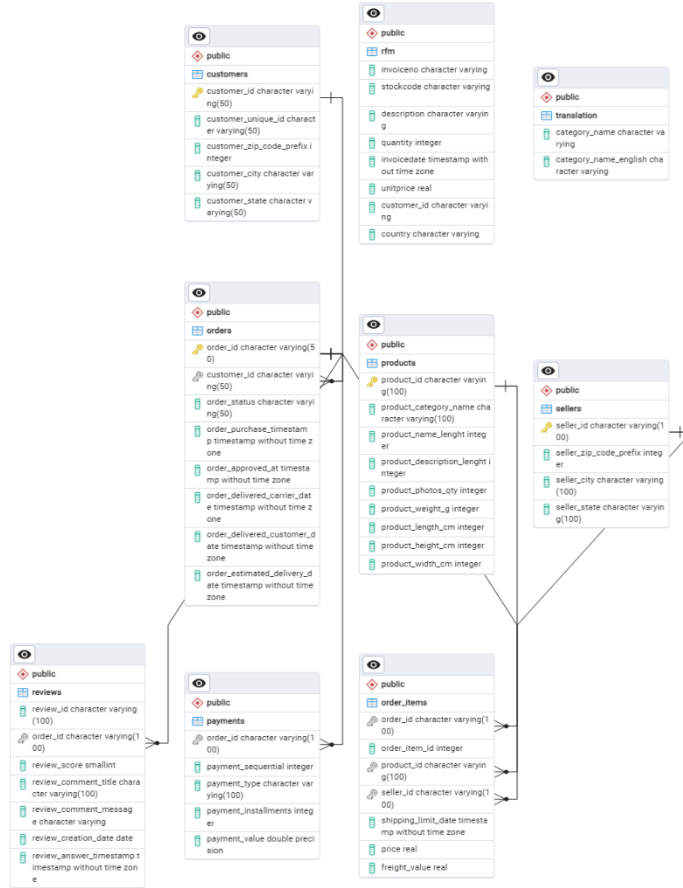
Brazilian E-Commerce Public Dataset by Olist | Kaggle

**GÖKÇE AKARSU**

## SQL PROJESİ TANITIMI

Aşağıda verilen linkteki veri setinden veri tabanını oluşturarak ERD oluşturulmuştur.

Link : [Brazilian E-Commerce Public Dataset by Olist | Kaggle](https://www.kaggle.com/olist-br/brazilian-e-commerce)



## Case 1 : Sipariş Analizi

Question 1 :

-Aylık olarak order dağılımını inceleyiniz. Tarih verisi için order\_approved\_at kullanılmalıdır.

```
SELECT      to_char(order_approved_at, 'YYYY-MM') AS date_y_m,  
            COUNT(DISTINCT order_id) AS order_count  
FROM orders  
WHERE order_approved_at IS NOT null  
GROUP BY 1  
ORDER BY 1;
```

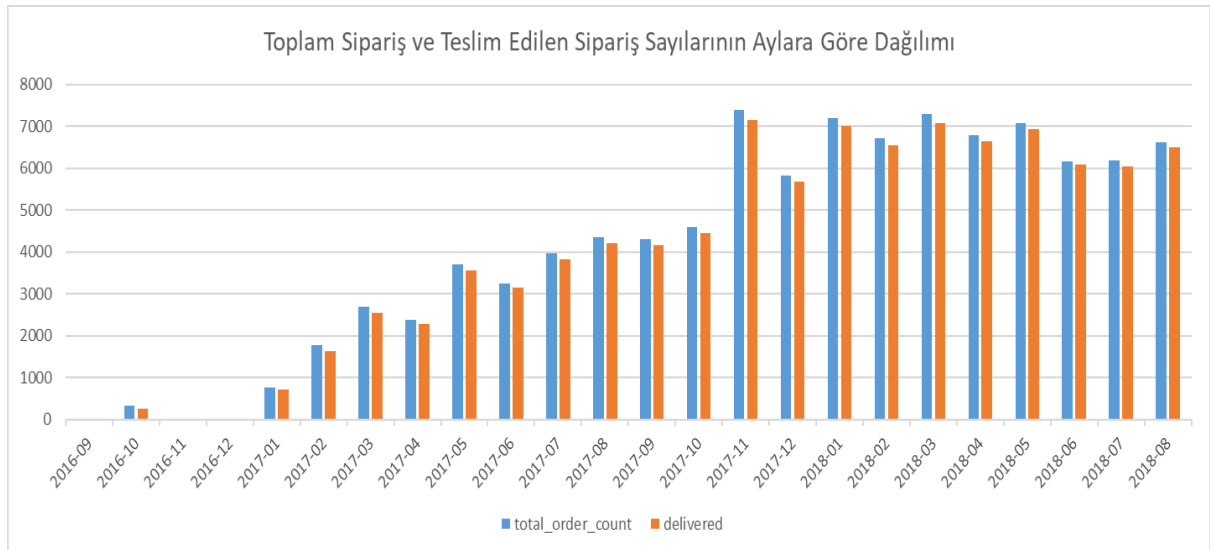


Aylık olarak order dağılımını ele aldığımız grafiğimizde 2017 yılı Ocak ayı itibariyle sipariş sayılarında artış görülmektedir. Bu artış 2017 Kasım ayında zirve yapmıştır. Kasım ayında sipariş sayılarının artmasının nedeninin Black Friday indirimlerinden ve yaklaşan Noel hazırlıklarından kaynaklandığını söyleyebiliriz. Aralık 2017 itibariyle sipariş sayısında az da olsa azalmalar meydana gelmiş ve yeni yıl ile birlikte 2018 Ocak ayında tekrar yükselişe geçmiştir. Sonraki aylarda ise sipariş sayılarının kendi trendini koruduğunu söyleyebiliriz.

## Question 2 :

-Aylık olarak order status kırılımında order sayılarını inceleyiniz. Sorgu sonucunda çıkan outputu excel ile görselleştiriniz. Dramatik bir düşüşün ya da yükselişin olduğu aylar var mı? Veriyi inceleyerek yorumlayınız.

```
SELECT      to_char(order_approved_at, 'YYYY-MM') as date_y_m,
            order_status,
            count(distinct order_id) as order_count
FROM orders
WHERE order_approved_at IS NOT null
GROUP BY 1,2
ORDER BY 1;
```

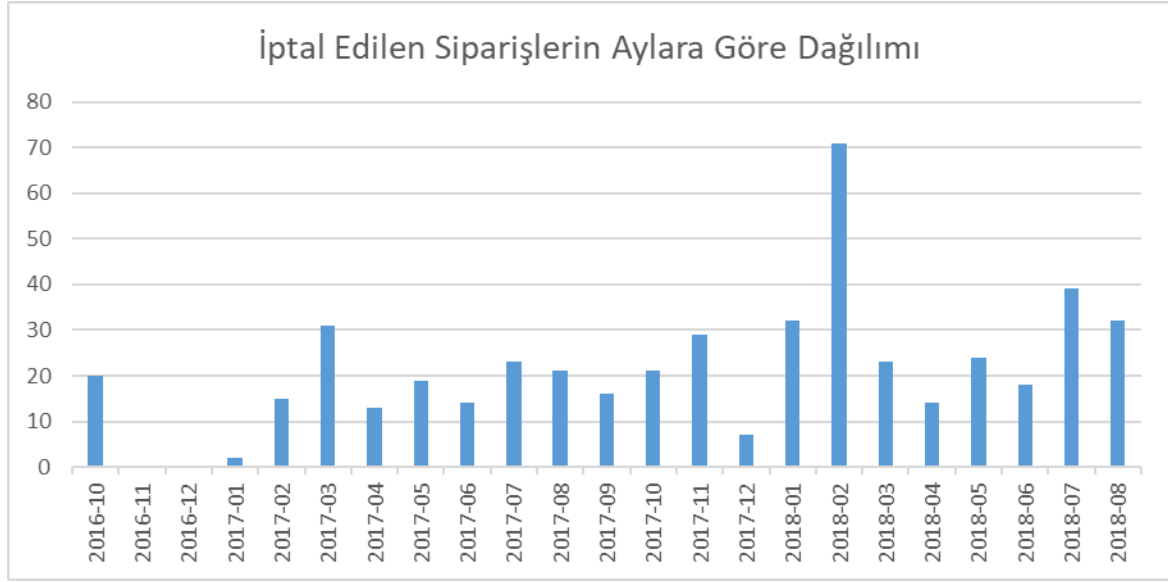


Bu soruda order status dağılımına geçmeden önce aylık olarak toplam verilen ve teslim edilen sipariş sayılarını incelediğimizde sipariş sayılarının paralel olarak ilerlediğini gözlemliyoruz.

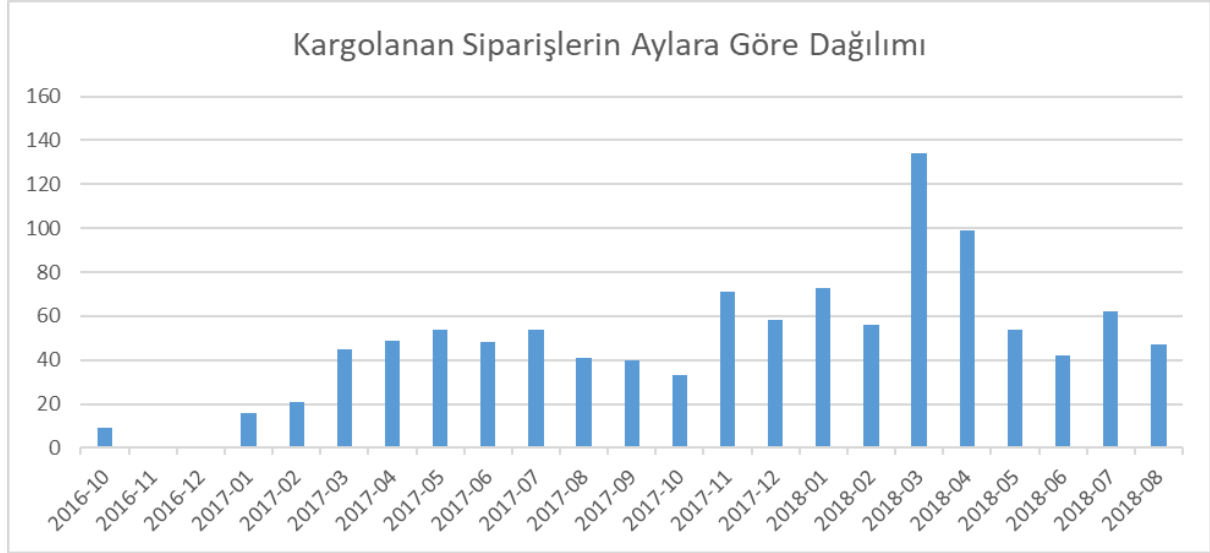
En yüksek sipariş teslimatı 2017 yılının Kasım ayında gerçekleşirken toplamda en çok siparişte bu ayda alınmıştır. Bu durumu hem Black Friday indirimlerine hem de yaklaşan Noel hazırlıklarına bağlayabiliriz.

2016 yılına ait verileri incelediğimizde sipariş sayılarının çok az olduğunu ve 2017 yılına bakarak bu verilerin anlamsız olduğunu söyleyebiliriz. Bu nedenle veri akışında ya bir problem olduğu ya da henüz müşteri veri tabanının yeni yeni oluştuğunu söyleyebiliriz.

2018 yılına ait verileri incelediğimizde ise sipariş sayılarının ay dağılımının kendi içinde tutarlı oldukça iyi olduğunu söyleyebiliriz. 2018 yılı Mart ayının teslim edilen sipariş sayısı bakımından diğer aylara göre daha yüksek olduğunu görmekteyiz. Bunun nedeninin 1 Nisan 2018 yılında kutlanan Paskalya öncesi hazırlıklar olduğunu söyleyebiliriz.



İptal edilen siparişleri incelediğimizde özellikle 2018 yılı şubat ayında rekor seviyeye ulaştığını görmekteyiz. Bu durumun şubat ayında gerçekleşen Bahia Karnavalı sebebiyle kargolarda yaşanan yoğunluk ve geç teslim tarihlerinden kaynaklandığını söyleyebiliriz.



2018 yılı mart ayında kargo sayılarında rekor bir seviye görmekteyiz. Bu bir önceki grafikte de bahsettiğimiz Bahia Karnavalı sebebiyle yaşanan aksaklıkların mart ayında düzene girmesiyle anlamlandırılabilir. Yine 2018 yılı 1 Nisan da kutlanmış olan Paskalya Bayramı da bu durumu olumlu şekilde etkilemiştir.

Question 3 :

-Ürün kategorisi kırılımında sipariş sayılarını inceleyiniz. Özel günlerde öne çıkan kategoriler nelerdir? Örneğin yılbaşı, sevgililer günü...

--Ürün kategori kırılımında teslim edilen siparişler--

```
SELECT      tr.category_name_english,
            COUNT(distinct o.order_id) AS order_count
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
JOIN products p ON oi.product_id = p.product_id
JOIN translation AS tr ON tr.category_name=p.product_category_name
WHERE o.order_status = 'delivered'
GROUP BY 1
ORDER BY 2 DESC;
```

-- 2018 yılbaşı noel --

```
SELECT      tr.category_name_english,
            COUNT(distinct o.order_id) AS order_count
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
JOIN products p ON oi.product_id = p.product_id
JOIN translation AS tr ON tr.category_name=p.product_category_name
WHERE o.order_status = 'delivered'
AND tr.category_name_english IS NOT NULL
AND o.order_purchase_timestamp BETWEEN '2017-12-01' AND '2018-01-01'
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

-- 2017 anneler günü--

```
SELECT      tr.category_name_english,
            COUNT(distinct o.order_id) AS order_count
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
JOIN products p ON oi.product_id = p.product_id
JOIN translation AS tr ON tr.category_name=p.product_category_name
WHERE o.order_status = 'delivered'
AND tr.category_name_english IS NOT NULL
AND o.order_purchase_timestamp BETWEEN '2017-04-14' AND '2017-05-14'
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

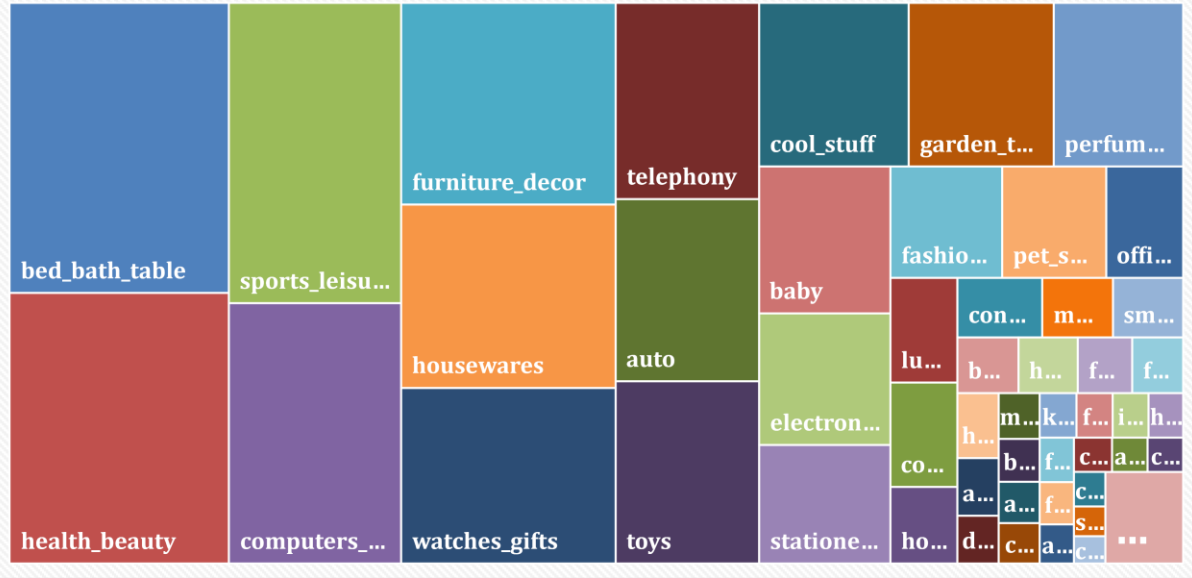
--2017 çocuklar günü—

```
SELECT      tr.category_name_english,
            COUNT(distinct o.order_id) AS order_count
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
JOIN products p ON oi.product_id = p.product_id
JOIN translation AS tr ON tr.category_name=p.product_category_name
WHERE o.order_status = 'delivered'
AND tr.category_name_english IS NOT NULL
AND o.order_purchase_timestamp BETWEEN '2017-09-12' AND '2017-10-12'
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

-- 2017 Black Friday—

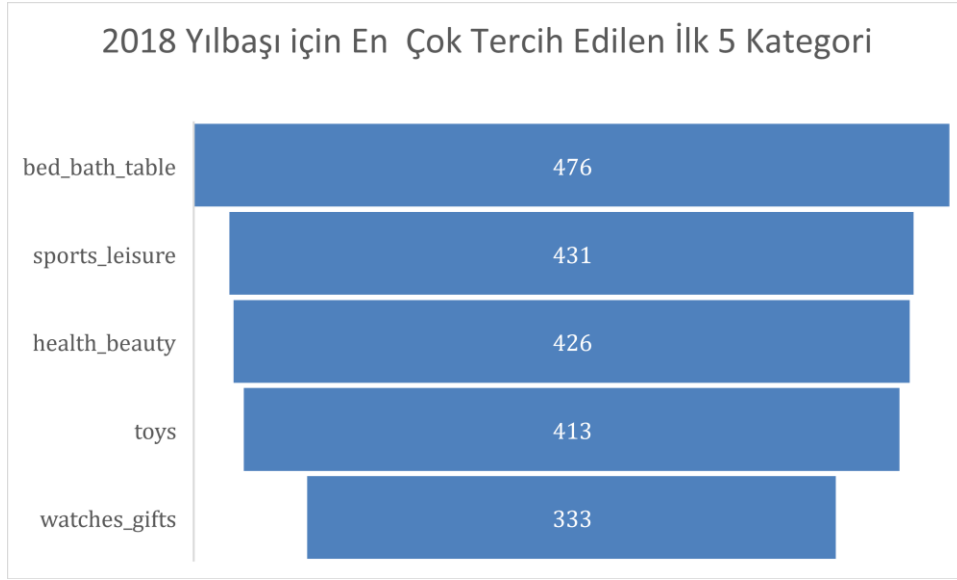
```
SELECT      tr.category_name_english,
            COUNT(distinct o.order_id) AS order_count
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
JOIN products p ON oi.product_id = p.product_id
JOIN translation AS tr ON tr.category_name=p.product_category_name
WHERE o.order_status = 'delivered'
AND tr.category_name_english IS NOT NULL
AND o.order_purchase_timestamp::date = '2017-11-24'
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

## Ürün Kategorisi Kırılımında Sipariş Sayıları

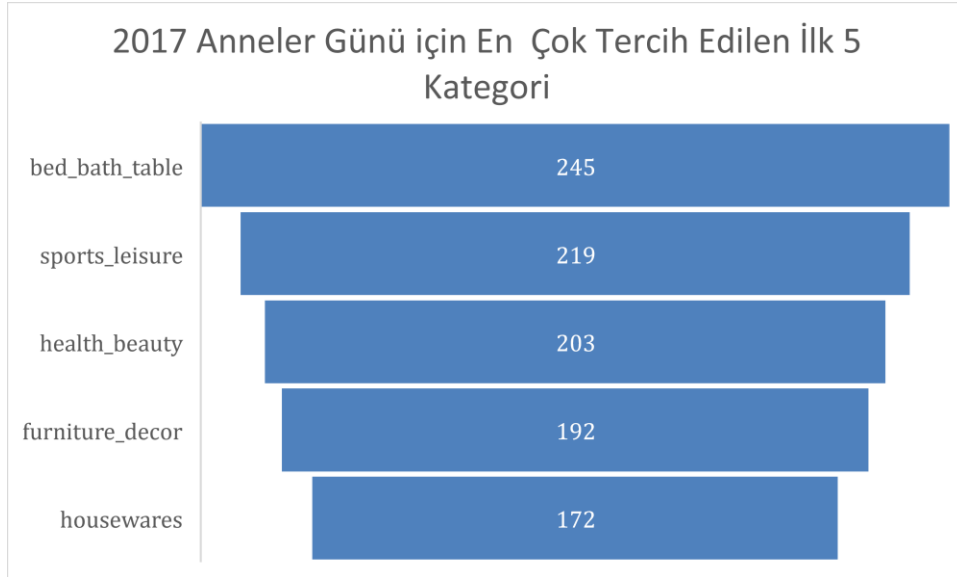


Bu soruda özellikle ele aldığımız kısım özel günler olduğu için teslim edilen siparişler üzerinden grafiklerimi oluşturmak istedim. Yukarıda verilen grafikte de görüldüğü gibi en çok tercih edilen Bed Bath Table mağazası olmuştur. Mağazayı incelediğimiz zaman içerik açısından çok zengin olduğunu görmekteyiz. Ev dekorasyonu, banyo ürünleri, mutfak ürünleri hatta çocuklar için oyuncaklar, nevresim takımları, havlular gibi çok çeşitli ürün yelpazesine sahiptir. 2. Sırada yer alan Health Beauty kişisel bakım, temizlik ürünleri gibi ürünler içermektedir. 3. Sırada ise Sports Leisure yer almaktadır.

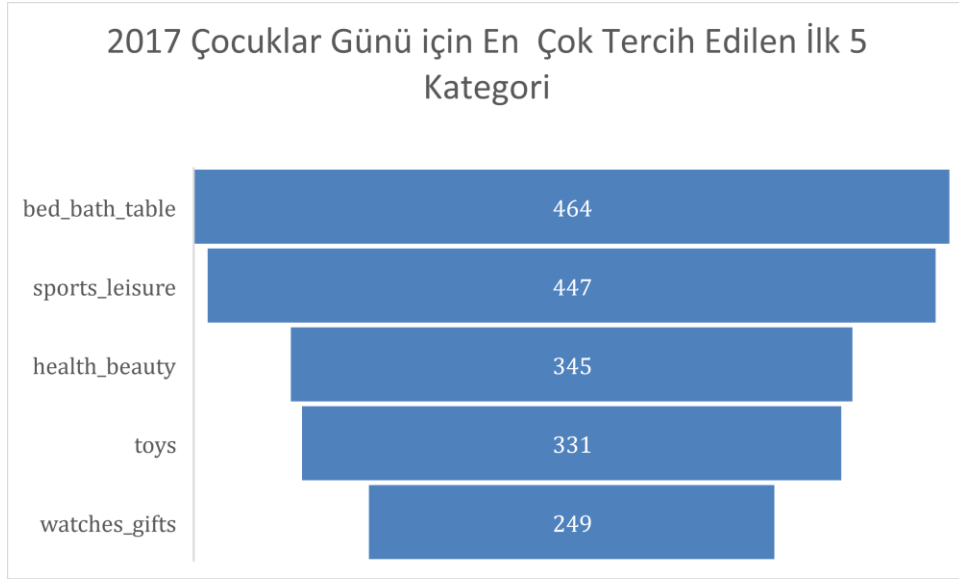




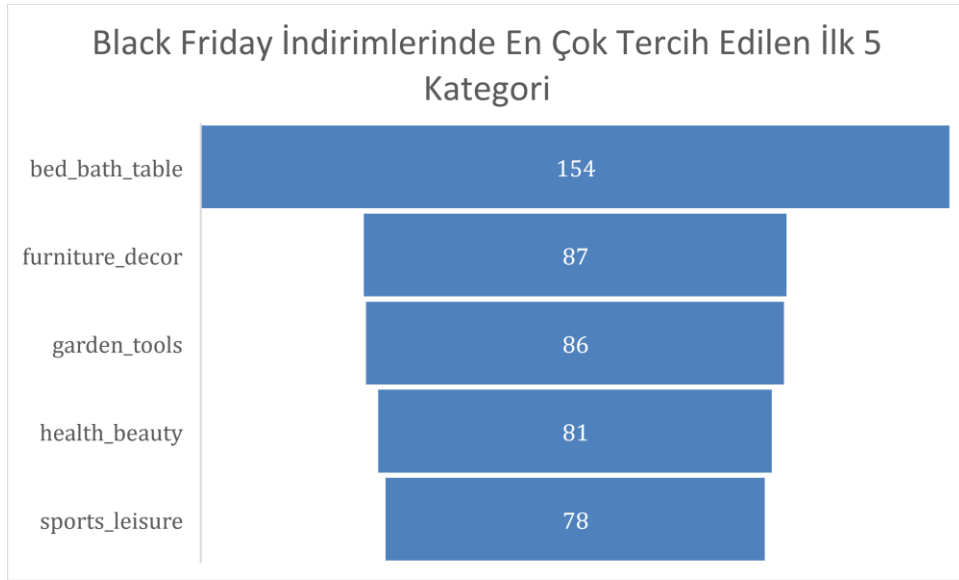
2018 yılı yılbaşı dönemi verilerini incelediğimiz zaman müşterilerin en çok tercih ettiği ürünler beklenildiği gibi ev dekorasyon ürünleri olmuştur. Bunun yanı sıra yine noel ve yılbaşının etkisinden dolayı oyuncak ve kol saati kategorilerinde özellikle bu dönemde artış olduğu görülmüştür.



Anneler gününe özel kampanyalar yaptığını düşündüğüm normalde de ülkede en çok tercih edilen Bed Bath Table mağazası yine ilk sırada yer almaktadır. Bunun dışında özellikle 2017 yılı için nisan mayıs ayları verileri incelendiğinde anneler günü sebebiyle mobilya dekorasyon(furniture decor) ve ev eşyaları (housewares) kategorisinde özellikle bu aylarda artış görülmüştür. Bu da beklenen ile uyumludur.



Çocuklar günü için veri setini incelediğimizde normalde de ülkede en çok satış yapan kategorilerin dışında listeye oyuncak ve saat eklenmiştir. Bu da beklenen ile uyumludur.



Black Friday indirimini incelediğimizde yine ülkede en çok satış yapan mağaza olan Bed Beth Table karşımıza çıkıyor. Bunun dışında mobilya dekorasyonu ve bahçe aletleri kategorilerinde ciddi bir artış görülmektedir. Buradan bu kategorilerde ciddi indirimler yapıldığını belirleyebiliriz.

Question 4 :

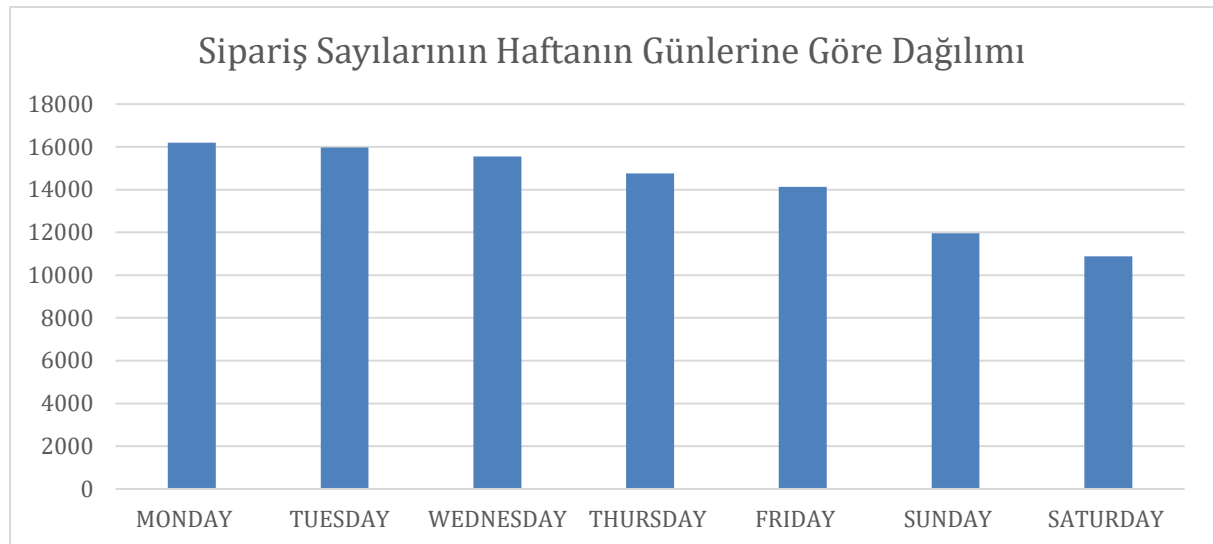
-Haftanın günleri(pazartesi, perşembe, ....) ve ay günleri (ayın 1'i,2'si gibi) bazında order sayılarını inceleyiniz. Yazdığınız sorgunun outputu ile excel'de bir görsel oluşturup yorumlayınız.

--haftanın günleri

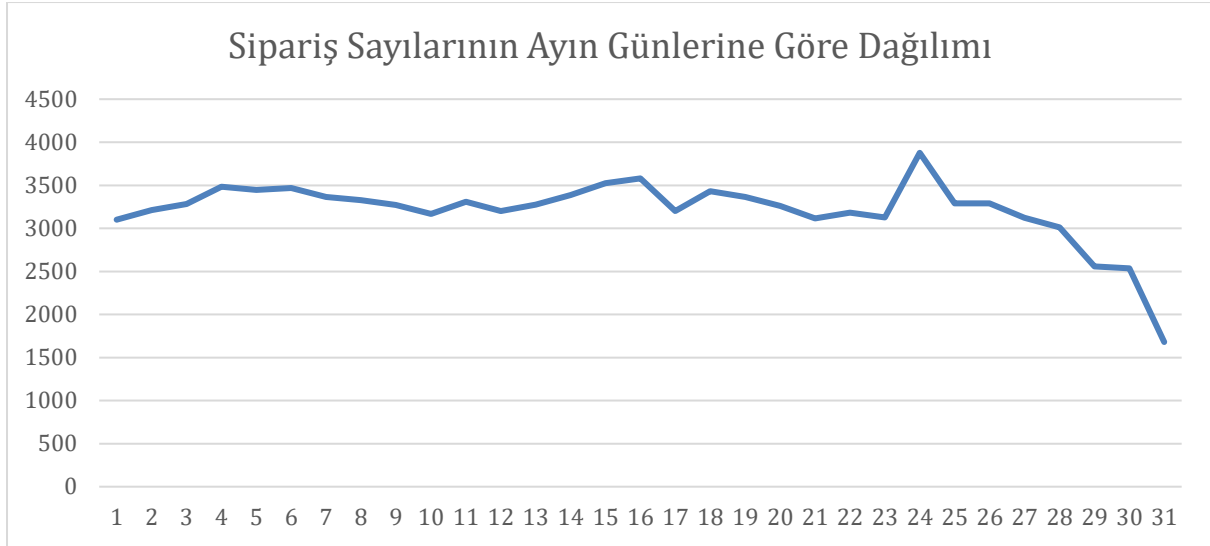
```
SELECT      to_char(order_purchase_timestamp, 'DAY') AS day_of_week,
            COUNT(DISTINCT order_id) AS order_count
FROM orders
GROUP BY 1
ORDER BY 2 DESC
;
```

--ayın günleri

```
SELECT      to_char(order_purchase_timestamp, 'DD') AS day_of_month,
            COUNT(DISTINCT order_id) AS order_count
FROM orders
GROUP BY 1
ORDER BY 1
;
```



Yukarıda verilen grafiği incelediğimizde en yüksek sayıda siparişin Pazartesi günü verildiğini, en düşük sipariş sayısının ise hafta sonu verildiğini görmekteyiz. Hafta sonu ülkenin genelinin dinlenmekte olduğunu veya etkinliklerde aktif olduğunu düşünürsek alışveriş yapmaya zaman ayırmamaları da beklenen bir durumdur. Haftanın başında ise ihtiyaçlara yönelik planlamalarla güne başlayan insanlar alışveriş konusunda da aktif rol almaktadır.



Ayın günlerine göre sipariş sayılarını incelediğimizde ayın sonunda ve başında düşen siparişler olduğunu görmekteyiz. Bunun yanında ayın 15-16'sı gibi ve ayın 24'ünde sipariş sayılarında artış yaşanmaktadır. Ülkede maaş ödemelerinin bu aylarda yapıldığını düşünebiliriz.

## Case 2 : Müşteri Analizi

Question 1 :

-Hangi şehirlerdeki müşteriler daha çok alışveriş yapıyor? Müşterinin şehrini en çok sipariş verdiği şehir olarak belirleyip analizi ona göre yapınız.

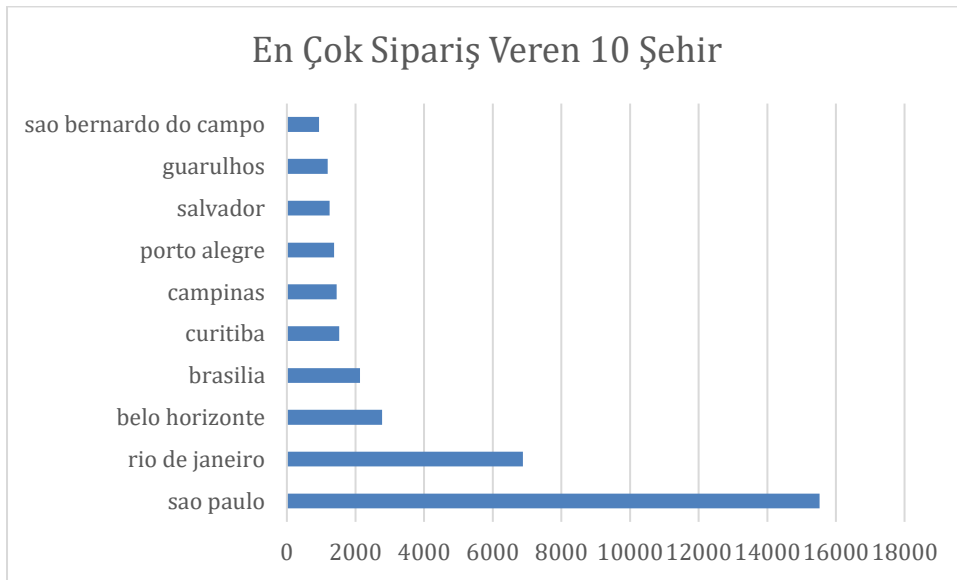
**Örneğin; Sibel Çanakkale'den 3, Muğla'dan 8 ve İstanbul'dan 10 sipariş olmak üzere 3 farklı şehirden sipariş veriyor. Sibel'in şehrini en çok sipariş verdiği şehir olan İstanbul olarak seçmelisiniz ve Sibel'in yaptığı siparişleri İstanbul'dan 21 sipariş vermiş şekilde görünmelidir.**

```
WITH customer_orders AS (  
  SELECT  
    cs.customer_unique_id,  
    cs.customer_city,  
    COUNT(DISTINCT o.order_id) AS order_count,  
    ROW_NUMBER() OVER(PARTITION BY cs.customer_unique_id ORDER BY  
COUNT(o.order_id) DESC) AS row_num  
  FROM customers AS cs  
  JOIN orders AS o ON o.customer_id = cs.customer_id  
  --WHERE customer_unique_id='f34cd7fd85a1f8baff886edf09567be3'  
  GROUP BY 1,2  
  ORDER BY 1  
)  
,  
customer_orders_2 AS (  
  SELECT  
    co.customer_unique_id,
```

```

        co.customer_city
    FROM customer_orders AS co
    WHERE co.row_num = 1
    GROUP BY 1,2
)
SELECT
    co2.customer_city,
    SUM(co.order_count) AS order_count
FROM customer_orders AS co
JOIN customer_orders_2 AS co2 ON co.customer_unique_id = co2.customer_unique_id
GROUP BY 1
ORDER BY 2 DESC;

```



En çok sipariş veren şehirler incelendiğinde özellikle ülkenin güneyinde yer alan şehirlerin ağırlıkta olduğunu görmekteyiz. İlk sırada yer alan Sao Paulo Brezilya, Amerika kıtası ve güney yarımküredeki en büyük şehirdir. Dünyanın dördüncü büyük metropoliten alanıdır. Dünyanın 11. En büyük şehridir. Brezilya ve Latin Amerika'daki en büyük ekonomiye sahiptir. İkinci sırada yer alan Rio De Janerio Brezilya'nın ikinci en büyük kentidir. Üçüncü sırada yer alan Belo Horizonte ise Brezilya'nın en büyük üçüncü şehridir. Tüm bunları göz önüne alırsak büyükşehirlerde nüfusun sosyoekonomik seviyesi daha yüksek ve internet kullanma alışkanlıkları daha fazla olduğu için sipariş sayıları daha fazladır diyebiliriz.

### Case 3: Satıcı Analizi

Question 1 :

-Siparişleri en hızlı şekilde müşterilere ulaştıran satıcılar kimlerdir? Top 5 getiriniz. Bu satıcıların order sayıları ile ürünlerindeki yorumlar ve puanlamaları inceleyiniz ve yorumlayınız.

--Öncelikte ortalama sipariş sayısı ve yorum sayısını hesaplayarak kendime bir kısıtlama belirledim.

WITH seller\_information AS

```
(
    SELECT
        s.seller_id,
        COUNT(o.order_id) AS order_count,
        AVG(o.order_delivered_customer_date - o.order_approved_at) AS avg_delivery_time,
        COUNT(r.review_comment_message) AS message_count
    FROM sellers AS s
    LEFT JOIN order_items AS oi ON s.seller_id = oi.seller_id
    LEFT JOIN orders AS o ON oi.order_id = o.order_id
    LEFT JOIN reviews AS r ON o.order_id = r.order_id
    WHERE order_delivered_customer_date IS NOT NULL AND order_approved_at IS NOT
    NULL
    GROUP BY 1
)
SELECT
    ROUND(AVG(order_count), 0) AS average_order_count,
    ROUND(AVG(message_count), 0) AS average_message_count
FROM
    seller_information;
```

--Ortalama Sipariş Sayısı 37 , Ortalama Yorum Sayısı 16'dır.

```
SELECT
    s.seller_id,
    ROUND(AVG(r.review_score),0) AS avg_review_score,
    COUNT(o.order_id) AS order_count,
    AVG(AGE(o.order_delivered_customer_date , o.order_approved_at))AS
avg_delivery_time,
    COUNT(r.review_comment_message) AS message_count
FROM sellers AS s
LEFT JOIN order_items AS oi ON s.seller_id=oi.seller_id
LEFT JOIN orders AS o ON oi.order_id=o.order_id
LEFT JOIN reviews AS r ON o.order_id=r.order_id
WHERE order_delivered_customer_date IS NOT NULL AND order_approved_at IS NOT
NULL
```

```

GROUP BY 1
HAVING COUNT(o.order_id)> 37
AND
COUNT(r.review_comment_message)>16
ORDER BY 4
LIMIT 5
;

```

	seller_id [PK] character varying (100)	avg_review_score numeric	order_count bigint	avg_delivery_time interval	message_count bigint
1	6b90f847357d8981edd79a1eb1bf0acb	4	48	4 days 23:41:09.520833	17
2	d566c37fa119d5e66c4e9052e83ee4ea	5	73	4 days 32:59:35.931507	23
3	138dbe45fc62f1e244378131a6801526	4	58	5 days 19:14:25.034483	24
4	289cdb325fb7e7f891c38608bf9e0962	5	125	5 days 29:01:08.864	49
5	45ba18c210d42734ec52c0c1c574e9ee	4	38	5 days 29:07:46.631579	18

Öncelikte bu soruda kendime bazı kısıtlamalar getirdim. Sipariş sayısının anlamlı bir yorumlama da kullanılabilmesi için ortalama sipariş sayılarını belirledim. Sipariş sayısı ortalamanın üzerinde olan satıcılar ve aynı şekilde yorum sayıları ortalama yorum yapan kişilerin sayısının üzerinde olan satıcıları analiz ettim. Yorumları incelediğimizde bu satıcılara ait yorumlarda hızlı teslimattan memnuniyetini belirttiğini görebiliriz. Yorumları incelemek için her bir seller\_id yi girerek aşağıdaki sorguyu kullanabiliriz.

```

--yorumlar
SELECT
    s.seller_id,
        r.review_comment_message
FROM sellers AS s
LEFT JOIN order_items AS oi ON s.seller_id = oi.seller_id
LEFT JOIN orders AS o ON oi.order_id = o.order_id
LEFT JOIN reviews AS r ON o.order_id = r.order_id
WHERE s.seller_id='289cdb325fb7e7f891c38608bf9e0962'
GROUP BY 1,2

```

Question 2 :

-Hangi satıcılar daha fazla kategoriye ait ürün satışı yapmaktadır?  
Fazla kategoriye sahip satıcıların order sayıları da fazla mı?

```

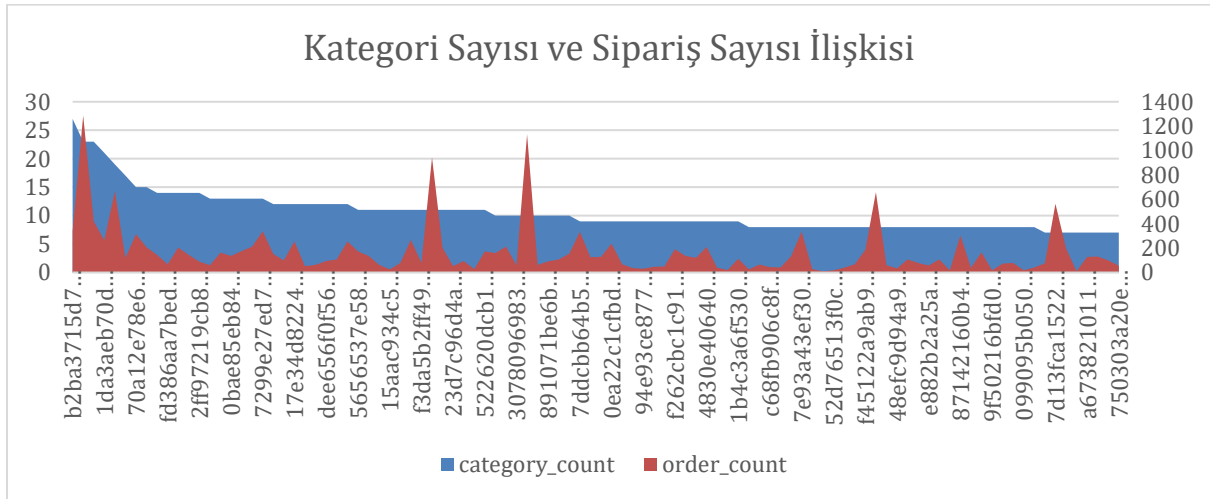
SELECT
    s.seller_id,
        COUNT(DISTINCT tr.category_name_english) AS category_count,
        COUNT(DISTINCT oi.order_id) AS order_count
FROM sellers AS s
LEFT JOIN order_items AS oi ON s.seller_id=oi.seller_id
LEFT JOIN products AS pr ON oi.product_id=pr.product_id
LEFT JOIN translation AS tr ON pr.product_category_name=tr.category_name
GROUP BY 1
ORDER BY 2 DESC

```

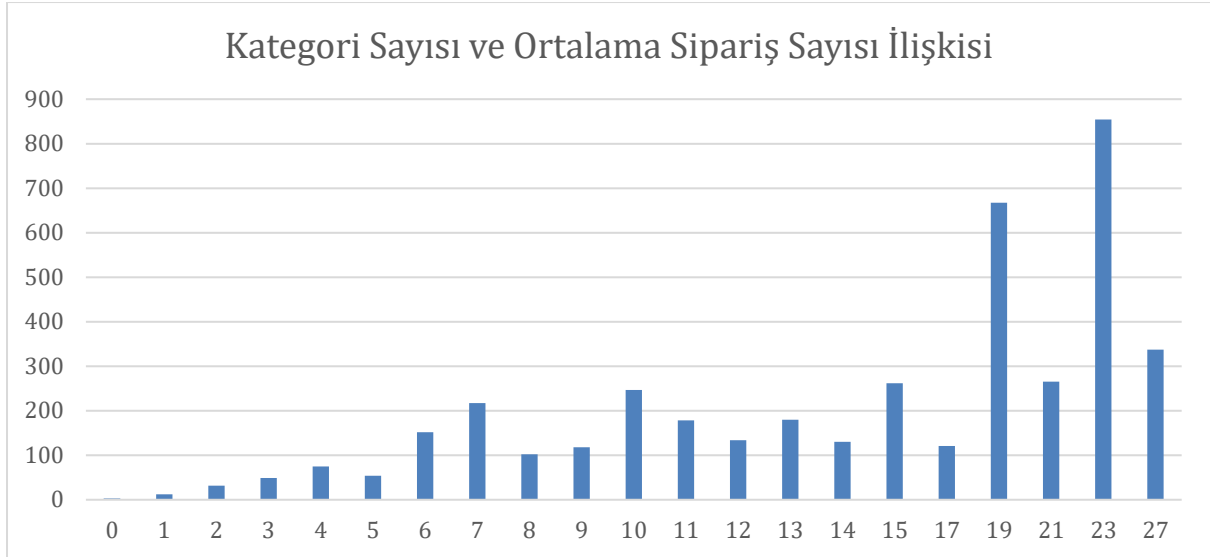
LIMIT 100;

--Kategori Sayısının Sipariş Sayısı Üzerine Etkisi

```
WITH seller_category_count AS (  
    SELECT  
        s.seller_id,  
        COUNT(DISTINCT tr.category_name_english) AS category_count,  
        COUNT(DISTINCT oi.order_id) AS order_count  
    FROM sellers AS s  
    LEFT JOIN order_items AS oi ON s.seller_id=oi.seller_id  
    LEFT JOIN products AS pr ON oi.product_id=pr.product_id  
    LEFT JOIN translation AS tr ON pr.product_category_name=tr.category_name  
    GROUP BY 1  
)  
SELECT  
    category_count,  
    ROUND(AVG(order_count)) AS avg_order_count  
FROM seller_category_count  
GROUP BY 1  
ORDER BY 1;
```







Kategori sayısı ve sipariş sayısı arasındaki ilişkiyi incelediğimizde daha az kategoriye sahip olup benzer sipariş sayılarına sahip olan satıcılar olduğunu görmekteyiz. Aynı şekilde kategori sayılarının ortalama sipariş sayılarına göre dizilimine baktığımızda da örneğin 7 kategoriye sahip bir satıcıların ortalama sipariş sayısı 217 iken 17 kategoriye sahip satıcıların ortalama sipariş sayısının 121 olduğunu görmekteyiz. Buradan geniş ürün yelpazesine sahip satıcıların büyüme stratejilerinin her zaman başarılı olamadığını daha az sayıda ürün satıp uzmanlaşmış satıcıların benzer sayıda sipariş alabildiği sonucuna varmaktayız.

#### Case 4 : Payment Analizi

Question 1 :

-Ödeme yaparken taksit sayısı fazla olan kullanıcılar en çok hangi bölgede yaşamaktadır?  
Bu çıktıyı yorumlayınız.

---en çok müşteriye ve siparişe sahip şehirler

```
SELECT
    cs.customer_state,
    cs.customer_city,
    COUNT(DISTINCT customer_unique_id) AS customer_count,
    COUNT(DISTINCT o.order_id) AS order_count
FROM customers AS cs
LEFT JOIN orders AS o ON o.customer_id=cs.customer_id
LEFT JOIN payments as py ON py.order_id=o.order_id
WHERE payment_type='credit_card' AND payment_installments>2
GROUP BY 1,2
HAVING COUNT(o.order_id)>10
ORDER BY 4 DESC;
```

---taksit sayıları-sipariş sayıları-müşteri sayıları ilişkisi

```
SELECT
  cs.customer_state,
  py.payment_installments,
  COUNT(DISTINCT customer_unique_id) AS customer_count,
  COUNT(DISTINCT o.order_id) AS order_count
FROM customers AS cs
LEFT JOIN orders AS o ON o.customer_id=cs.customer_id
LEFT JOIN payments as py ON py.order_id=o.order_id
WHERE payment_type='credit_card' AND payment_installments>2
GROUP BY 1,2
HAVING COUNT(o.order_id)>10
ORDER BY 2 DESC;
```

	customer_state character varying (50)	customer_city character varying (50)	customer_count bigint	order_count bigint
1	SP	sao paulo	4858	5024
2	RJ	rio de janeiro	2612	2689
3	MG	belo horizonte	1099	1136
4	DF	brasilia	746	766
5	BA	salvador	556	571
6	RS	porto alegre	533	558
7	PR	curitiba	528	542
8	SP	campinas	478	493
9	SP	guarulhos	422	427
10	RJ	niteroi	313	324

En çok sipariş alan bölgelere baktığımız zaman ülkenin en büyük üç şehri olan Sao Paulo, Rio De Janeiro ve Belo Horizonte en yüksek sipariş sayılarıyla karşımıza çıkmaktadır. En yüksek siparişe sahip bölgeler yukarıdaki tabloda görülmektedir. Bunlar SP, RJ, MG, DF, BA, RS, PR'dir.

Şimdi taksit sayılarına göre müşterilerin en çok hangi bölgede yaşadıklarına bakalım.

	customer_state character varying (50)	payment_installments integer	customer_count bigint	order_count bigint
1	SP	15	22	24
2	MG	12	11	11
3	RJ	12	21	21
4	RS	12	13	13
5	SP	12	39	39

Özellikle 12 taksit ve üzerindeki bölgeleri ele almak istedim. Yukarıdaki tabloda da görüldüğü gibi 12 taksit ve üzerinde alışveriş yapılan bölgeler genellikle ülkenin güneyinde bulunan ve

ülkenin sosyoekonomik olarak önde gelen büyükşehirlerini barındıran bölgelerinden oluşmaktadır. Bu bölgeler eğitim ve refah seviyesi yüksek, ekonomik olarak dünya geneliyle yarışan ve satın alma gücü yüksek bölgelerdir. Yine bu bölgelerde internet kullanıcısının daha fazla olması da bu durumu etkileyen faktörlerdendir.

Question 2 :

-Ödeme tipine göre başarılı order sayısı ve toplam başarılı ödeme tutarını hesaplayınız. En çok kullanılan ödeme tipinden en az olana göre sıralayınız.

```
SELECT
    DISTINCT py.payment_type,
    COUNT(o.order_id)OVER (PARTITION BY payment_type) AS suc_order_count,
    ROUND(SUM(py.payment_value)OVER (PARTITION BY payment_type)) AS total_payment
FROM payments AS py
LEFT JOIN orders AS o ON o.order_id = py.order_id
WHERE o.order_status = 'delivered'
ORDER BY 3 DESC;
```

	payment_type character varying (100) 🔒	suc_order_count bigint 🔒	total_payment double precision 🔒
1	credit_card	74586	12101095
2	boleto	19191	2769933
3	voucher	5493	343013
4	debit_card	1486	208421

Question 3 :

-Tek çekimde ve taksitle ödenen siparişlerin kategori bazlı analizini yapınız. En çok hangi kategorilerde taksitle ödeme kullanılmaktadır?

--Tek Çekim-Peşin Ödeme

```
SELECT
    tr.category_name_engliSh,
    COUNT(DISTINCT o.order_id)
FROM payments AS py
LEFT JOIN order_items AS oi ON oi.order_id = py.order_id
LEFT JOIN orders AS o ON o.order_id = oi.order_id
LEFT JOIN products AS pr ON pr.product_id=oi.product_id
LEFT JOIN translation AS tr ON tr.category_name=pr.product_category_name
WHERE py.payment_installments = 1
AND tr.category_name_engliSh IS NOT NULL
GROUP BY 1
ORDER BY 2 DESC;
```

## --Taksitli Ödeme

```
SELECT      tr.category_name_engliSh,  
            COUNT(DISTINCT o.order_id)  
FROM payments AS py  
LEFT JOIN order_items AS oi ON oi.order_id = py.order_id  
LEFT JOIN orders AS o ON o.order_id = oi.order_id  
LEFT JOIN products AS pr ON pr.product_id=oi.product_id  
LEFT JOIN translation AS tr ON tr.category_name=pr.product_category_name  
WHERE py.payment_installments > 1  
AND tr.category_name_engliSh IS NOT NULL  
GROUP BY 1  
ORDER BY 2 DESC;
```

	category_name_english character varying	count bigint
1	sports_leisure	4299
2	computers_accessories	4177
3	health_beauty	3880
4	bed_bath_table	3535
5	furniture_decor	3152
6	housewares	2732
7	telephony	2552
8	electronics	2052
9	auto	2041
10	toys	1902

Yukarıda verilen tabloda tek çekim ile yapılan ödemelerin sipariş sayılarına göre sıralamasını görmekteyiz. En çok tercih edilen kategorilerin spor, bilgisayar ve aksesuar, sağlık ve güzellik, dekorasyon ve ev ürünleri, teknolojik ürünler, otomobil ve oyuncak olduğunu görmekteyiz.

	category_name_english character varying	count bigint
1	bed_bath_table	5965
2	health_beauty	5006
3	watches_gifts	3794
4	sports_leisure	3480
5	furniture_decor	3353
6	housewares	3197
7	computers_accessories	2562
8	cool_stuff	2217
9	toys	2008
10	perfumery	1945

Yukarıda verilen tabloda taksitli ödemeleri görmekteyiz. Taksitli ödemelerde ise en çok tercih edilen yine ülke genelinde de çok sevilen Bed Bath Table mağazası olduğunu görmekteyiz. Büyük ve pahalı ürünler almak isteyen müşteriler taksitli ödeme seçeneği sayesinde bu mağazayı tercih etmektedir. Bunun dışında yine tek çekim tablosundan farklı olarak saat ve parfüm alışverişleri öne çıkmaktadır. Diğer kategoriler tek çekim ile de alınan kategoriler ile benzerdir.

### Case 5 : RFM Analizi

Aşağıdaki e\_commerce\_data\_.csv dosyasındaki veri setini kullanarak RFM analizi yapınız. Recency hesaplarırken bugünün tarihi değil en son sipariş tarihini baz alınız.

Veri seti bu linkten alınmıştır, veriyi tanımak için linke girip inceleyebilirsiniz.

#### [E-Commerce Data](#)

```
SELECT MAX(invoicedate::date) FROM rfm; --2011-12-09
```

```
--Recency
```

```
SELECT
    customer_id,
    MAX(invoicedate::date) AS max_invoicedate,
    '2011-12-09'::date - MAX(invoicedate::date) AS recency
FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 3;
```

--Frequency

```
SELECT
    customer_id,
    COUNT(DISTINCT invoiceno) AS frequency
FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 2 DESC;
```

--Monetary

```
SELECT
    customer_id,
    ROUND(SUM(quantity*unitprice)::numeric) AS monetary
FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 2 DESC;
```

--R-F-M Analizi

```
WITH recency AS
(
    SELECT
        customer_id,
        MAX(invoicedate::date) AS max_invoicedate,
        '2011-12-09'::date - MAX(invoicedate::date) AS recency
    FROM rfm
    WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
    GROUP BY 1
    ORDER BY 3
),
frequency AS
(
    SELECT
        customer_id,
        COUNT(DISTINCT invoiceno) AS frequency
    FROM rfm
    WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
    GROUP BY 1
    ORDER BY 2 DESC
),
monetary AS
(
    SELECT
        customer_id,
        ROUND(SUM(quantity*unitprice)::numeric) AS monetary
```

```

FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 2 DESC
)
SELECT
    r.customer_id,
    r.recency,
    NTILE(5) OVER (ORDER BY recency DESC) AS receny_score,
    f.frequency,
    CASE WHEN f.frequency>=1 AND f.frequency<=4
    THEN f.frequency
    ELSE 5 END AS frequeny_score,
    m.monetary,
    NTILE(5) OVER (ORDER BY monetary)AS monetary_score
FROM recency AS r
JOIN frequency AS f ON r.customer_id=f.customer_id
JOIN monetary AS m ON r.customer_id=m.customer_id
ORDER BY f.frequency DESC;

```

--r-f-m score table---

```

WITH rfm_score_table_2 AS
(
    WITH rfm_score_table AS
    (
        WITH recency AS
        (
            SELECT
            customer_id,
            MAX(invoicedate::date) AS max_invoicedate,
            '2011-12-09'::date - MAX(invoicedate::date) AS recency
FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 3
),
frequency AS
(
SELECT
    customer_id,
    COUNT(DISTINCT invoiceno) AS frequency
FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 2 DESC
),

```

```

monetary AS
(
SELECT
    customer_id,
    ROUND(SUM(quantity*unitprice)::numeric) AS monetary
FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 2 DESC
)
SELECT
    r.customer_id,
    r.recency,
    NTILE(5) OVER (ORDER BY recency DESC) AS recency_score,
    f.frequency,
    CASE WHEN f.frequency>=1 AND f.frequency<=4
    THEN f.frequency
    ELSE 5 END AS frequency_score,
    m.monetary,
    NTILE(5) OVER (ORDER BY monetary)AS monetary_score
FROM recency AS r
JOIN frequency AS f ON r.customer_id=f.customer_id
JOIN monetary AS m ON r.customer_id=m.customer_id
ORDER BY f.frequency DESC
)
SELECT
    rfm.customer_id,
    rfm.recency_score::text || '-' || rfm.frequency_score::text || '-' || rfm.monetary_score::text as
rfm_score
FROM rfm_score_table AS rfm
)
SELECT
    rst.rfm_score,
    COUNT(rst.customer_id) AS count_customer
FROM rfm_score_table_2 AS rst
GROUP BY 1
ORDER BY 2 DESC;

```

--r-f score table---

```

WITH rfm_score_table_2 AS
(
    WITH rfm_score_table AS
    (
        WITH recency AS
        (
            SELECT
                customer_id,

```



```

        MAX(invoicedate::date) AS max_invoicedate,
        '2011-12-09'::date - MAX(invoicedate::date) AS recency
FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 3
),
frequency AS
(
SELECT
    customer_id,
    COUNT(DISTINCT invoiceno) AS frequency
FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 2 DESC
),
monetary AS
(
SELECT
    customer_id,
    ROUND(SUM(quantity*unitprice)::numeric) AS monetary
FROM rfm
WHERE customer_id IS NOT NULL and invoiceno NOT LIKE 'C%'
GROUP BY 1
ORDER BY 2 DESC
)
SELECT
    r.customer_id,
    r.recency,
    NTILE(5) OVER (ORDER BY recency DESC) AS recency_score,
    f.frequency,
    CASE WHEN f.frequency>=1 AND f.frequency<=4
    THEN f.frequency
    ELSE 5 END AS frequency_score,
    m.monetary,
    NTILE(5) OVER (ORDER BY monetary)AS monetary_score
FROM recency AS r
JOIN frequency AS f ON r.customer_id=f.customer_id
JOIN monetary AS m ON r.customer_id=m.customer_id
ORDER BY f.frequency DESC
)
SELECT
    rfm.customer_id,
    rfm.recency_score::text || '-' || rfm.frequency_score::text as rf_score
FROM rfm_score_table AS rfm
)
SELECT

```

```

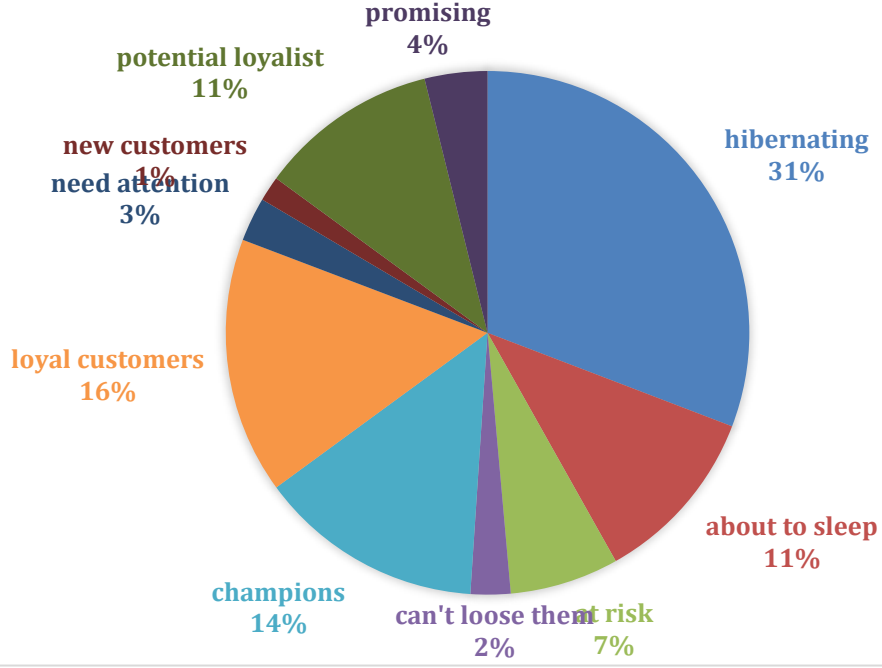
rst.rf_score,
COUNT(rst.customer_id) AS count_customer
FROM rfm_score_table_2 AS rst
GROUP BY 1
ORDER BY 2 DESC;

```



	rf_score text 🔒	count_customer bigint 🔒
1	1-1	624
2	5-5	506
3	2-1	341
4	4-5	320
5	3-1	295
6	2-2	216
7	3-2	184
8	3-5	182
9	4-1	167
10	4-2	158
11	1-2	157
12	2-3	137
13	4-3	126
14	5-2	120
15	3-3	117
16	5-4	97
17	4-4	97
18	3-4	90
19	2-4	88
20	2-5	86
21	5-3	77
22	5-1	67
23	1-3	51
24	1-5	21
25	1-4	15

## MÜŞTERİLERİN RFM SINIFLARINA GÖRE DAĞILIMI



Sorgu sonucunda müşterileri recency ve frequency değerlerine göre sınıflandırdım. Toplam müşterilerin sınıflara göre yüzdesel dağılımı yukarıdaki grafikteki gibidir. Grafiğin büyük çoğunluğu %31 ile hibernating müşterilerden oluşmaktadır. Bu müşteriler için kampanyalar yapıp daha çok alışveriş yapmaları konusunda teşvik edilebilir.